Dokumentacja Specyfikacji Wymagań (SRS)

Projekt: Pełna Analiza Text Mining jednego pliku (przetwarzanie tekstu i analiza sentymentu z wykorzystaniem słowników w plikach CSV)
Wersja dokumentu: 1.0
Data: 25.03.2025
Autor: [Anna Kowalska, Adam Kowalski]

1. Wprowadzenie:

Niniejszy dokument opisuje specyfikację wymagań dla skryptu R, który realizuje pełną analizę text mining i analizę sentymentu na podstawie zawartości pliku .txt. System łączy tradycyjne techniki czyszczenia tekstu, tokenizację, stemming oraz uzupełnienie rdzeni słów po stemmingu (typ prevalent) z oceną sentymentu przy użyciu słowników w plikach CSV (AFINN, Bing, NRC, Loughran), a także słowników z pakietu SentimentAnalysis (GI, HE, LM, QDAP). Dodatkowo generowane są wizualizacje częstości słów w postaci chmury słów, wykresów rodzaju sentymentu wg słowników oraz wykresów zmiany sentymentu w czasie.

2. Cele systemu:

- Wczytanie tekstu wejściowego (plik .txt) z odpowiednim kodowaniem (UTF-8)
- Przetwarzanie i oczyszczanie tekstu (normalizacja, tokenizacja, stemming)
- Usunięcie nieistotnych słów (stopwords)
- Zliczenie częstości występowania słów oraz ich wizualizacja w formie chmury
- Przeprowadzenie analizy sentymentu z użyciem słowników:
    - w plikach CSV (AFINN, Bing, NRC, Loughran)
    - wbudowanych w pakiet SentimentAnalysis (GI, HE, LM, QDAP)
- Wizualizacja wyników sentymentu za pomocą wykresów słupkowych i czasowych
- Porównanie wyników sentymentu między słownikami
- Umożliwienie analizy zmian sentymentu w czasie

3. Wymagania funkcjonalne:

- **Wczytywanie danych**:
    - Skrypt powinien umożliwiać wczytanie danych tekstowych z lokalnego pliku .txt.
    - Skrypt powinien obsługiwać kodowanie UTF-8.
- **Przetwarzanie i oczyszczanie tekstu:**
    - Skrypt powinien umożliwiać normalizację apostrofów na apostrof klasyczny.
    - Skrypt powinien umożliwiać usunięcie liczb, interpunkcji oraz form skróconych zawierających apostrofy.
    - Skrypt powinien umożliwiać usunięcie pustych elementów oraz zbędnych znaków specjalnych.
    - Skrypt powinien umożliwiać usunięcie stopwords z pakietów tidytext i tm.
    - Skrypt powinien umożliwiać wykonanie stemmingu i uzupełnienia rdzeni słów.

- **Analiza częstości:**
  - Skrypt powinien umożliwiać zliczenie liczby wystąpień słów.
  - Skrypt powinien umożliwiać posortowanie słów według częstości.
  - Skrypt powinien umożliwiać przedstawienie wyników w formie tabeli i chmury słów.
- **Analiza sentymentu (słowniki w plikach CSV):**
  - Skrypt powinien umożliwiać wczytanie słowników: afinn.csv, bing.csv, nrc.csv, loughran.csv.
  - Skrypt powinien umożliwiać dopasowanie słów do słowników i zliczenie sentymentów.
  - Skrypt powinien umożliwiać filtrowanie słów o sentymencie pozytywnym lub negatywnym.
- **Analiza sentymentu (słowniki wbudowane w pakiet SentimentAnalysis):**
  - Skrypt powinien przeprowadzać analizę sentymentu tekstu z wykorzystaniem biblioteki SentimentAnalysis.
  - Skrypt powinien wykorzystywać słowniki GI, HE, LM, QDAP.
  - Skrypt powinien konwertować ciągłe wartości sentymentu na wartości kierunkowe.
  - Skrypt powinien umożliwiać podział tekstu na segmenty o ustalonej długości.
- **Wizualizacja danych**:
  - Skrypt powinien umożliwiać wizualizację wyników (wykresy ggplot2).
  - Skrypt powinien generować wykresy skumulowanego sentymentu dla każdego słownika.
  - Skrypt powinien generować wykres porównujący sentyment na podstawie różnych słowników.
  - Skrypt powinien generować wykresy przedstawiające ewolucję sentymentu w czasie (wykresy liniowe i wygładzone).
- **Agregacja danych**:
  - Skrypt powinien agregować sentyment z różnych słowników w jednej ramce danych.
  - Skrypt powinien usuwać brakujące wartości (NA).

4. Wymagania niefunkcjonalne:

- **Wydajność**:
  - Analiza pliku o długości 1000 zdań powinna trwać nie dłużej niż 15 sekund.
- **Bezpieczeństwo:**
  - System powinien zapewnić poprawność danych wyjściowych.
- **Niezawodność**:
  - Skrypt powinien poprawnie obsługiwać różne formaty danych tekstowych.
  - Skrypt powinien poprawnie obsługiwać brakujące wartości.
- **Użyteczność**:
  - Wykresy powinny być czytelne i zawierać odpowiednie etykiety.
  - Skrypt powinien umożliwiać wykonanie wizualizacji z użyciem ggplot2 i motywu theme_gdocs dla lepszej czytelności.
  - Skrypt powinien umożliwiać generowanie chmury słów z wykorzystaniem kolorystyki RColorBrewer.

- **Kompatybilność**:
    - Skrypt powinien być kompatybilny z R w wersji 4.0 lub nowszej.
    - Skrypt powinien korzystać z bibliotek tm, tidytext, stringr, ggplot2, ggthemes, SentimentAnalysis, SnowballC, tidyverse..

5. Interfejsy użytkownika:

- **Wejście**:
    - Plik tekstowy .txt.
    - Pliki słowników w formacie .csv.
- **Wyjście**:
    - Tabela z częstością występowania słów.
    - Chmura słów (wordcloud).
    - Wykresy słupkowe rodzaju sentymentu wg słowników (AFINN, Bing, NRC, Loughran, GI, HE, LM, QDAP).
    - Wykresy zmian sentymentu w czasie (liniowe i wygładzone).

6. Wymagania dotyczące danych:

- Skrypt zakłada, że dane tekstowe są w języku angielskim.
- Skrypt nie obsługuje analizy sentymentu dla innych języków.
- Skrypt wykorzystuje słowniki sentymentów dostępne w plikach .CSV oraz w pakiecie SentimentAnalysis.
- Skrypt nie obsługuje analizy sentymentu dla danych tekstowych z innych źródeł niż pliki .txt.
- Skrypt nie obsługuje plików o rozmiarze powyżej 100 MB.

Słownictwo dokumentacji:

- **Token**: pojedynczy element tekstu (słowo).
- **Stopwords**: słowa niewnoszące wartości semantycznej do analizy.
- **Sentyment**: emocjonalne nastawienie w tekście.
- **Słownik sentymentów**: lista słów i ich ocen wg sentymentu.
- **Skumulowany sentyment**: suma ocen sentymentu dla całego tekstu.
- **Wartości kierunkowe**: konwersja ciągłych wartości sentymentu na kategorie (np. pozytywny, negatywny, neutralny).
- **Ewolucja sentymentu**: zmiana sentymentu w czasie (wzdłuż czasu narracyjnego).
- **Stem**: forma słowa po sprowadzeniu go do rdzenia.
- **Stem Completion**: uzupełnienie rdzenia słowa po stemmingu.

Przypadki użycia (use cases)

- Użytkownik:

- wczytuje plik .txt.
- uruchamia analizę
- wyświetla wyniki
- generuje wykresy i raport html

- Skrypt/system:

- przetwarza tekst
- oczyszcza tekst
- analizuje sentyment tekstu przy użyciu słowników
- generuje chmurę słów
- generuje wykresy skumulowanego sentymentu
- generuje wykres porównujący rodzaj sentymentu wg słowników
- generuje wykresy zmiany sentymentu w czasie narracyjnym

Testowe przypadki użycia:

- Test z plikiem .txt zawierającym tekst o pozytywnym sentymencie.
- Test z plikiem .txt zawierającym tekst o negatywnym sentymencie.
- Test z plikiem .txt zawierającym tekst o neutralnym sentymencie.
- Test z plikiem .txt zawierającym tekst o mieszanym sentymencie.
- Test z plikiem .txt zawierającym brakujące wartości.
- Test z plikiem .txt zawierającym znaki specjalne.

Scenariusze użytkownika (user stories)

Scenariusz 1: Analiza opinii klientów o produkcie

- **Jako:** Analityk marketingowy
- **Chcę:** Przeanalizować opinie klientów o nowym produkcie z pliku tekstowego
- **Aby:** Zrozumieć ogólny sentyment klientów i zidentyfikować obszary, które wymagają poprawy.

Kryteria akceptacji:

- Użytkownik może wczytać plik tekstowy z opiniami klientów.
- Skrypt przeprowadza analizę sentymentu za pomocą różnych słowników.
- Skrypt generuje wykresy skumulowanego sentymentu i porównuje wyniki z różnych słowników.
- Skrypt generuje wykresy ewolucji sentymentu w czasie.
- Użytkownik może zidentyfikować ogólny sentyment klientów i obszary, które wymagają poprawy.

Scenariusz 2: Monitorowanie sentymentu w mediach społecznościowych

- **Jako:** Specjalista ds. mediów społecznościowych
- **Chcę:** Monitorować sentyment w mediach społecznościowych
- **Aby:** Reagować na negatywne opinie i wzmacniać pozytywny wizerunek marki.

Kryteria akceptacji:

- Użytkownik może wczytać dane tekstowe z mediów społecznościowych (np. z Twittera) do pliku tekstowego.
- Skrypt przeprowadza analizę sentymentu i generuje wykresy ewolucji sentymentu w czasie.
- Użytkownik może monitorować zmiany sentymentu.
- Użytkownik może identyfikować nagłe zmiany sentymentu i reagować na nie.
- Użytkownik może generować raporty z analizy sentymentu.

Scenariusz 3: Analiza przemówień

- **Jako:** Analityk ekonomii politycznej
- **Chcę:** Przeanalizować przemówienie w celu określenia dominujących emocji
- **Aby:** Zidentyfikować wpływ narracji na odbiorców.

Kryteria akceptacji:

- Użytkownik może wczytać plik tekstowy z treścią przemówienia.
- Skrypt przeprowadza analizę sentymentu za pomocą różnych słowników.
- Skrypt generuje chmurę słów.
- Skrypt generuje wykresy skumulowanego sentymentu i porównuje wyniki z różnych słowników.
- Skrypt generuje wykresy ewolucji sentymentu w czasie.
- Użytkownik może zidentyfikować sentymenty dominujące w narracji przemówienia w celu dokonania pogłębionej analizy naukowej.